MTR and RLL Constraints with Unconstrained Positions

T. Lei Poo
Marvell Semiconductor Inc.
5488 Marvell Lane
Santa Clara CA 95054, USA
Email: lpoo@marvell.com

Brian H. Marcus
Department of Mathematics
University of British Columbia
Vancouver BC V6T 1Z2, Canada
Email: marcus@math.ubc.ca

***Abstract*— In most recording channels, modulation codes are employed to transform user data to sequences that satisfy some desirable constraint. Run-length-limited (RLL) and maximum transition run (MTR) systems are examples of constraints that improve timing and detection performance. When used in conjunction with error correction codes (ECC), schemes that facilitate easy access to soft information for ECC decoding are preferred. In one such scheme introduced by Immink and Wijngaarden, certain bit positions in the modulation code are deliberately left unconstrained for the insertion of ECC parity bits. The overall code rate is a function of the density of unconstrained positions, called the tradeoff function. In our previous work with Chaichanavong, we presented properties of the tradeoff function, but exact closed form expressions were only known for three constraints. The present work[1] adds to the list of MTR and RLL constraints for which the tradeoff function is known exactly.**

I. Introduction

In recording channels, an error correcting code (ECC) and a modulation code are often used to improve the detection performance. An ECC improves the minimum distance, while a modulation code imposes a constraint on the recorded sequences. Some well-known binary constraints include the runlength-limited (RLL(d,k)) system, which limits the run of 0 to be at least d and at most k, and the maximum transition run (MTR(j,k)) [2], which limits the run of 0 to be at most k and the run of 1 to be at most j. When there is no restriction on the runs of 0, we say that $k=\infty$ and, by tradition, such a constraint is denoted by MTR(j) [5]. The binary sequences that satisfy the RLL(d,k) constraint (respectively, the MTR(j) constraint) can be obtained from consecutive edge labels of the graph in Fig. 1 (respectively, Fig. 2).

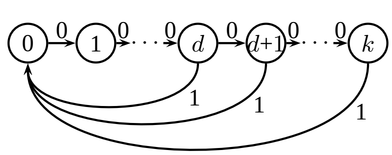

Fig. 1. A presentation for RLL(d,k) constraint

Typically, the ECC and the modulation code are concatenated so that the ECC is the outer code and the modulation code is the inner code. Since the modulation decoder is typically a

[1]This work was part of T.L. Poo's PhD Dissertation at the Electrical Engineering Department at Stanford University, Stanford CA

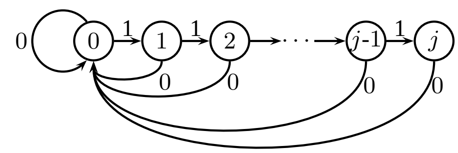

Fig. 2. A presentation for MTR(j) constraint

hard decoder, the passing of soft information from the channel to the ECC decoder is thus hindered, and as a result, the error correction capability is degraded. There have been several schemes proposed to overcome this limitation, such as soft decoding for modulation codes and reverse concatenation [3]. However, in this work, we focus on the scheme presented by Wijngaarden and Immink [8], and Campello et al [1], in which the modulation code is designed so that some pre-specified positions are "unconstrained." Such a position can take on any symbol without violating the constraint. Thus we can employ a systematic ECC with the appropriate rate and stuff the parity bits to these positions. The receiver can then decode the ECC first, without loss of soft information.

Example 1 ([8], [1]): Consider a block code for MTR(2) consisting of two codewords: $\{10101, 01101\}$. Let $0,1,2,3,4$ be the bit positions. If bits 2 or 4 of either codeword is flipped, the constraint is still satisfied for all concatenations of the codewords. Hence, we can map a user bit to $\{100, 010\}$ and then encode these words by a systematic rate-3/5 ECC (with bit positions 2 and 4 used for parity) to obtain an overall rate of 1/5. We say that the period is 5 and the insertion rate is 2/5. The unconstrained set is defined to be the set of bit positions that can be flipped, which is $\{2,4\}$.

The ECC design is beyond the scope of this paper. We are interested in finding analytical solutions to the maximum overall code rate f_S for a given constraint S and insertion rate. The overall code rate is a function of the insertion rate, called the *tradeoff function*. A lower bound to the tradeoff function is the code rate for a given period and unconstrained set. This is easily computed using the method in Campello et al [1]. In that work, exact closed form expressions for the tradeoff functions for MTR(1) and MTR(2) constraints are derived; see Fig. 3 (the tradeoff function for MTR(1) is a straight line, but for MTR(2) it is piecwise linear with a subtle kink at insertion rate $\rho = 1/3$).

Both tradeoff functions are achieved by timesharing of standard bit-stuffing schemes. Recall from [1] that a *b-bit-stuffing* (or

$b : j+1$ *bit-stuffing*) for MTR(j) begins with a string s of length N that satisfies the MTR($j-b$) constraint; then s is subdivided into intervals of size $j-b+1$ before a string of b ones is inserted in between each of these intervals. The resulting string satisfies MTR(j); the scheme has insertion rate $\rho = b/(j+1)$ and code rate equal to $(j-b+1)/(j+1)$ of the capacity of MTR($j-b$).

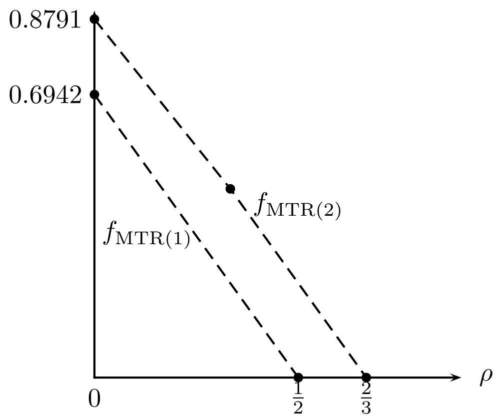

Fig. 3. Tradeoff Functions for MTR(1) and MTR(2).

In our previous work with Chaichanvong [7], we introduced a graph construction $\hat{G}$ on which the tradeoff function is defined, and which captures all information on valid periods and unconstrained sets for a given constraint. Using this graph construction, we derived the tradeoff function for MTR$(2,2)$, see Fig. 4.

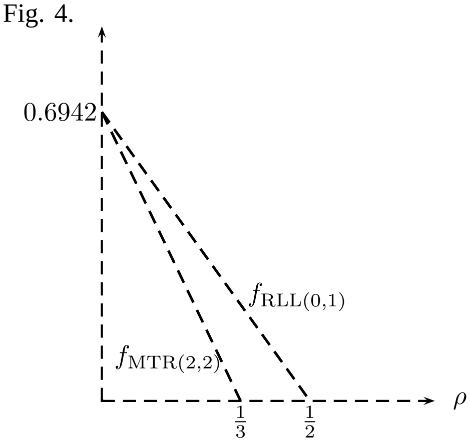

Fig. 4. Tradeoff functions for RLL$(0,1)$ and MTR$(2,2)$.

In the sections that follow, we first give a short background on constrained coding and on the tradeoff function. Then we establish the tradeoff function for RLL$(d, 2d+1)$. Finally, we show that for low insertion rates, the tradeoff function for MTR(3) is a straight line achieved by bit-stuffing schemes. Complete proofs can be found in [6].

II. Background on Constrained Coding

A *labeled graph* $G = (V, E, L)$ consists of

- a finite set of states $V = V_G$;
- a finite set of edges $E = E_G$, where each edge e has an *initial state* $\sigma(e)$ and a *terminal state* $\tau(e)$, both in V;
- an edge labeling $L = L_G : E \to \Sigma$, where Σ is a finite alphabet.

Formally, a *constrained system* or *constraint* $S = S(G)$ is the set of finite sequences obtained by reading the edge labels of a path in a labeled graph G. Such a graph is called a *presentation* of the constraint. An element in S is called a *word*. We use wz to denote the concatenation of two words w and z; $|w|$ denotes the length of w; w_i denotes the ith symbol in $w = w_0 w_1 \ldots w_{|w|-1}$.

We say G is *deterministic* if at each state, all outgoing edges carry distinct labels, and *irreducible* if for any pair (u, v) of states, there is a path from u to v. A reducible graph is a graph consisting of many irreducible subgraphs, called the *irreducible components* with possible *transitions* from component to component. A constrained system is said to be irreducible if it has an irreducible presentation. For an irreducible constraint, there is a unique minimal (in terms of the number of states) deterministic presentation, called the *Shannon cover*.

A labeled graph has *finite memory* if there is an integer m such that all paths of length m with the same labeling terminate at the same state. The smallest m for which this holds is called the *memory* of the graph. A constrained system is *finite-type* or has *finite memory* m if its Shannon cover has memory m. Many practical constrained systems, including RLL and MTR, are irreducible and finite-type.

The *follower set* of a state u in G, denoted by $\mathcal{F}(u) = \mathcal{F}_G(u)$, is defined to be the set of all finite words that can be generated from u in G. We allow x to be the empty word ϵ, in which case the follower set is all of S. Note that if x does not occur in S, then $\mathcal{F}(x)$ is empty.

For a graph G, the *adjacency matrix* $A = A_G$ is the $|V_G| \times |V_G|$ matrix whose entries are indexed by the states of G and $A_{u,v}$ is the number of edges from u to v in G.

The *capacity* of a constraint S, denoted cap_S, is defined as

$$\mathrm{cap}_S = \lim_{q\to\infty} \frac{1}{q} \log N(q; S),$$

where $N(q; S)$ is the number of words of length q in S. It is known that $\mathrm{cap}(S) = \log \lambda(A)$, where $\lambda(A)$ is the largest positive eigenvalue of the adjacency matrix A of a deterministic presentation of S. Refer to [4] for more details on constrained systems.

III. Tradeoff Function of a Constraint

Let $\Box$ denote an unconstrained symbol. This symbol represents the unconstrained position which is allowed to take on any symbol in the alphabet of the constraint.

Let $\mathcal{A}_2$ and $\mathcal{A}_3$ denote the alphabets $\{0,1\}$ and $\{0,1,\Box\}$ respectively. For an alphabet $\mathcal{A}$, define $\mathcal{A}^*$ to be the set of all finite words over $\mathcal{A}$. For a given word $w \in \mathcal{A}_3^*$, define

$$\begin{aligned}\Phi(w) \;\; = \;\; & \{x \in \mathcal{A}_2^* \; : \; |x| = |w|, \; x_i = w_i \text{ if } w_i = 0 \text{ or } 1, \\ & x_i \in \{0,1\} \text{ if } w_i = \Box\}\,.\end{aligned}$$

For example, if $w = 0\Box1\Box$ then $\Phi(w)$ has 4 *fillings*: $\{0010, 0011, 0110, 0111\}$.

Let S be a binary constrained system. Define $\hat{S} = \{w \in \mathcal{A}_3^* : \Phi(w) \subseteq S\}$. This set $\hat{S}$ is a collection of words w such that if we replace each occurrence $\Box$ independently by 0 or 1, we obtain a word in S. We showed previously in [7] that $\hat{S}$ is a constrained system presented by a graph $\hat{G}$, that is defined as follows.

- **States:** All intersections of the follower sets of words in S.
- **Transitions:** The transitions are given by

$$\bigcap_{i=1}^{k}\mathcal{F}(x_i)\xrightarrow{0}\bigcap_{i=1}^{k}\mathcal{F}(x_i0) \qquad \text{if } x_i0\in S \text{ for all } 1\le i\le k$$

$$\bigcap_{i=1}^{k}\mathcal{F}(x_i)\xrightarrow{1}\bigcap_{i=1}^{k}\mathcal{F}(x_i1) \qquad \text{if } x_i1\in S \text{ for all } 1\le i\le k$$

$$\bigcap_{i=1}^{k}\mathcal{F}(x_i)\xrightarrow{\Box}\bigcap_{b=0}^{1}\bigcap_{i=1}^{k}\mathcal{F}(x_ib) \qquad \text{if } x_i0, x_i1\in S \text{ for all } 1\le i\le k$$

An example using $\text{RLL}(0,1)$ is shown in Fig. 5.

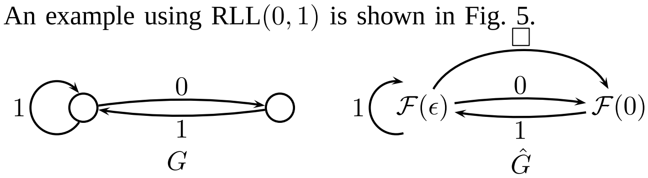

Fig. 5. The Shannon cover G and $\hat{G}$ for the $\text{RLL}(0,1)$

Let $I\subset\mathbb{N}$ and S be a constrained system. Define $N(q,I)$ to be the number of words $w\in\hat{S}$ such that

- $|w|=q$,
- $w_i=\Box$ if and only if $i\in I$.

Hence the set I represents the positions of $\Box$ in w; we say that w *satisfies* I.

For $0\le\rho\le 1$, define $\mathcal{I}(\rho)$ to be the set of all sequences (I_q) such that $I_q\subseteq\{1,2,\ldots,q\}$ and $\lim_{q\to\infty}|I_q|/q=\rho$.

A set $I\subseteq\mathbb{N}$ is said to be *periodic* if there exists a period $N\in\mathbb{N}$ and $U\subseteq\{1,\cdots,N\}$ such that $i\in I$ if and only if $i\equiv i' \mod N$ for some $i'\in U$. A sequence $(I_q)\in\mathcal{I}(\rho)$ is *periodic* if

- $I_q\subseteq I_{q+1}$ for all $q\in\mathbb{N}$, i.e., the sequence is increasing,
- $I=\bigcup_{q=1}^{\infty}I_q$ is a periodic set.

Let $\mathcal{I}_p(\rho)=\{(I_q)\in\mathcal{I}(\rho) \; : \; (I_q) \text{ is periodic}\}$. We define the maximum achievable code rate for periodic unconstrained positions as

$$f_S(\rho)=\max\left\{\sup_{(I_q)\in\mathcal{I}_p(\rho)}\limsup_{q\to\infty}\frac{\log N(q,I_q)}{q},0\right\}.$$

and call this the *tradeoff function*. Correspondingly, the maximum insertion rate is given by $\mu=\mu_S=\sup_{f_S(\rho)>0}\rho$.

Note that f_S is defined only on the rationals $0\le\rho\le 1$. When S is finite-type, this agrees with other definitions previously given (see [7, Theorem 44]). While f_S is defined directly on the set of sequences, it can be defined in terms of the graph $\hat{G}$. When S is finite-type, there is a single non-trivial irreducible component G' of $\hat{G}$ that contains all information needed to evaluate f_S; see [7, Corollary 27].

IV. Tradeoff Function for $\text{RLL}(d,2d+1)$

Let $S=\text{RLL}(d,2d{+}1)$. From [7, Theorem 25], the maximum insertion rate is $\mu_S=1/(2d+2)$. Clearly, an achievable rate region is given by the area under the straight line connecting the points $(0,\text{cap}_S)$ and $(1/(2d+2),0)$ since this can be achieved by timesharing of bit insertion schemes. It is shown in the next proposition that this line is indeed f_S.

Theorem 2: The tradeoff function $f_{\text{RLL}(d,2d+1)}$ is the line connecting $(0,\text{cap}_S)$ and $(1/(2d+2),0)$.

Sketch of proof: The main tool for this proof a matrix inequality to imply an eigenvalue inequality: namely, if X and Y are nonnegative matrices and $X\succcurlyeq Y$, where $\succcurlyeq$ denotes entry-by-entry inequality, then $\lambda(X)\ge\lambda(Y)$.

Since S is finite-type, it suffices to use the irreducible component G' of $\hat{G}$ (mentioned above) to compute f_S. For $\text{RLL}(d,2d{+}1)$, G' is shown in Fig. 6. Note that an unconstrained position with label $\Box$ is always followed by d constrained positions to reach state $\mathcal{F}(10^{2d+1})$. This can only happen in the follower set transition given below:

$$\mathcal{F}(10^d)\xrightarrow{\Box}\mathcal{F}(1)\cap\mathcal{F}(10^{d+1})\xrightarrow{0}\cdots\xrightarrow{0}$$
$$\mathcal{F}(10^{d-1})\cap\mathcal{F}(10^{2d})\xrightarrow{0}\mathcal{F}(10^{2d+1}).$$

Even though there are box edges emanating from $\mathcal{F}(10^s)$ for $s=d+1,d+2,\ldots,2d$ in $\hat{G}$, all these follower set transitions terminate in $\{\epsilon\}$ as shown below:

$$\mathcal{F}(10^s)\xrightarrow{\Box}\mathcal{F}(1)\cap\mathcal{F}(10^{s+1})\xrightarrow{0}\cdots\xrightarrow{0}$$
$$\mathcal{F}(10^{2d-s})\cap\mathcal{F}(10^{2d+1})\xrightarrow{\Box}\{\epsilon\}.$$

Order the states in the adjacency matrix $A_{G'}$ as follows:$\{\mathcal{F}(1),\cdots,\mathcal{F}(10^{2d+1}),\mathcal{F}(1)\cap\mathcal{F}(10^{d+1}),\cdots,\mathcal{F}(10^{d-1})\cap\mathcal{F}(10^{2d})\}$. Let A (B) denote the part of $A_{G'}$ corresponding to the edges with binary (box) labels.

Given a length N, a rational insertion rate $\rho\in[0,1/(2d{+}2)]$, and a set of unconstrained positions $U\subseteq\{0,\ldots,N-1\}$ with $|U|/N=\rho$, define the matrix representing the periodic (U,N) configuration by $M=M_0M_1\cdots M_{N-1}$, where $M_i=B$ if $i\in U$ and $M_i=A$ if $i\notin U$, with the indices taken mod N. Each entry of M represents the number of fillings in $\hat{S}$ satisfying U, with length exactly N, that begin and end with a restricted set of prefixes and suffixes respectively.

Since each B is followed by $2d+1$ A's, M can be viewed as a product of A's and C's, where $C=BA^{2d+1}$. It is easy to check that the matrix C has only one nonzero entry, namely a '1' at the diagonal entry value corresponding to the state $\mathcal{F}(10^d)$. In particular, C represents a periodic (U,N) configuration with insertion rate $1/(2d{+}2)$ and asymptotic code rate $\log_2(\lambda(C))/(2d+2)$ (which when evaluated equals 0).

Thus, for any N and $M=A^{n_1}C^{m_1}A^{n_2}C^{m_2}\ldots A^{n_l}C^{m_k}$, where n_i,m_i,l,k are nonnegative integers such that $\sum_{i=1}^{l}n_i+(2d{+}2)\sum_{i=1}^{k}m_i=N$, we can use the matrix relations $AC\preccurlyeq A$ and $C^n=C$ for any $n\ge1$ to upper bound M repeatedly:

$$\begin{aligned}M &= A^{n_1}C^{m_1}A^{n_2}C^{m_2}\ldots A^{n_l}C^{m_k},\\ &\preccurlyeq A^{n_1+n_2}C^{m_2}\ldots A^{n_l}C^{m_k},\\ &= A^{N-(2d+2)|U|}C^{|U|}=M',\end{aligned}$$

that spans the same number of time steps as M. The resulting matrix M' represents a timesharing scheme between the periodic configurations induced by C and A. ■

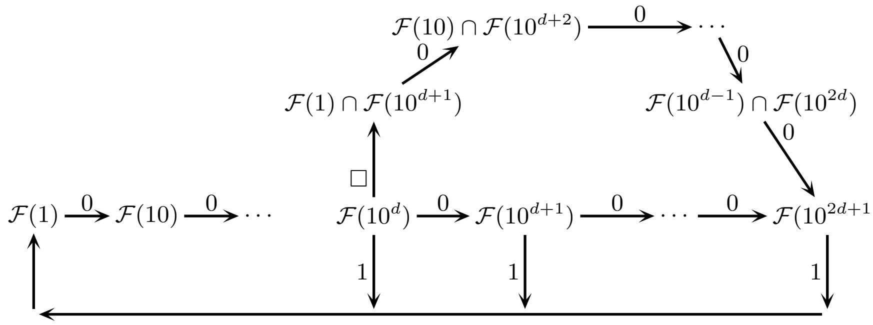

Fig. 6. The irreducible component G' of $\hat{G}$ for RLL$(d, 2d+1)$.

The tradeoff function for RLL$(d, 2d+1)$ is shown in Fig. 7.

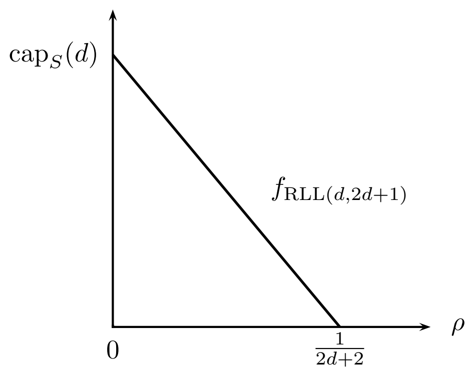

Fig. 7. Tradeoff function for RLL$(d, 2d+1)$.

V. Optimal Code Rates for MTR(j) at Low Insertion Rates

In this section, we describe results on the optimal configurations and code rates obtained for low insertion rates for MTR(j). From extensive computer computations on code rates of MTR(j) constraints in the low insertion rate region, we observed that high code rates are usually obtained from periodic configurations that "spread" the unconstrained positions apart by j spaces; this is in contrast to moderate and high insertion rates obtained from periodic configurations that cluster the unconstrained positions(see [1]).

Throughout this section, let $S = \mathrm{MTR}(j)$. The $\hat{G}$ presentation for S is given in Fig. 8. Let A (B) denote the part of the $(j+1) \times (j+1)$ adjacency matrix $A_{\hat{G}}$ corresponding to the edges with binary (box) labels. Let $C = BA^j$.

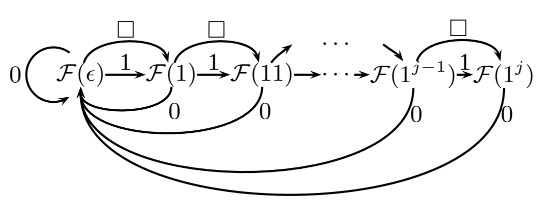

Fig. 8. The $\hat{G}$ presentation for MTR(j).

We are interested in the effect that the spacings between two unconstrained positions have on the code rates achieved. We refer to the spacing or the size of the spacing as a *gap* or *separation*, and use the terms interchangeably.

The main result of this section shows that the tradeoff function for MTR(3) is achieved by the bit-stuffing lower bound [1], [7] for insertion rates close to 0.

Theorem 3: Let $j = 3$. Let $g_1, \ldots, g_k \geq 0$ be gaps and $G = \sum_{i=1}^{k} g_i$. If

$$\frac{k}{k+G} < \frac{1}{3(j+1)},$$

then

$$\lambda(C^k A^{G-kj}) \geq \lambda(BA^{g_1} \cdots BA^{g_k}).$$

Note that the matrix C represents the 1-bit-stuffing scheme and is thus a periodic configuration that yields the second "corner point" in the standard bit-stuffing lower bound $h(\rho)$. Therefore, the matrix product $C^k A^{G-kj}$ can be viewed as a time-sharing scheme using 0-bit-stuffing and $1 : j+1$ bit-stuffing. Using this and concavity arguments, the tradeoff function for MTR(3) can be extended to the insertion rate region $[0, 1/(j+1)]$ as is stated in the following corollary.

Corollary 4: The graph of the tradeoff function $f_{\mathrm{MTR}(3)}(\rho)$ for $0 \leq \rho \leq 1/4$ is the straight line that connects $(0, \mathrm{cap}_{\mathrm{MTR}(3)})$ to $(1/4, (3/4)\mathrm{cap}_{\mathrm{MTR}(2)})$. So, $f_{\mathrm{MTR}(3)}(\rho)$ agrees with the standard bit-stuffing lower bound in this interval, shown in Fig. 9.

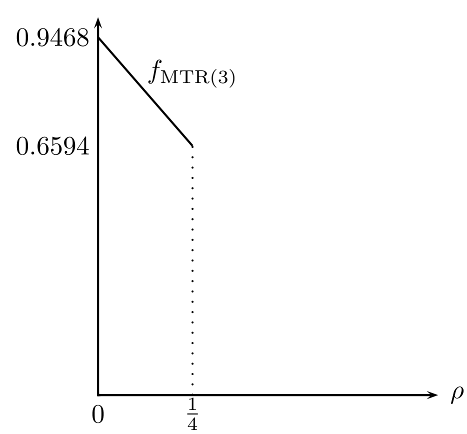

Fig. 9. Tradeoff Function for MTR(3) for $\rho \leq 1/4$.

We conjecture a version of Theorem 3 to hold for all j; this would imply:

Conjecture 5: The tradeoff function for MTR(j) for insertion rate $0 \leq \rho \leq \frac{1}{j+1}$ is a straight line given by the standard bit-stuffing lower bound $h(\rho)$.

Conjecture 5 holds true for MTR(1), MTR(2) and MTR(3). We have partial results in the direction of a proof for all j.

A. Topping up Small Gaps

We say a gap g_i is *large* if $g_i \geq j$ and *small* otherwise. The next lemma states that for MTR(3), any stretch of small gaps $(< j)$ can be topped up to size j simultaneously no matter how long the stretch is.

Lemma 6 (Small Separation Lemma for MTR(3)): Let $j = 3$. Let $g_1, \ldots, g_{k-1} \geq 0$ be small gaps and let $G = \sum_{i=1}^{k-1} g_i$. Then

$$A^{j-g_{k-1}} C^{k-1} BA^{j+1} \succcurlyeq \\ A^{j-g_{k-1}} BA^{g_{k-1}} BA^{g_{k-2}} \cdots BA^{g_1} BA^{kj-G+1}.$$

B. Reduction of Large Gaps

When the period is a multiple of $(j+1)$, the average gap size is exactly j, so the resulting configuration with uniformly spaced unconstrained positions corresponds to the $1 : j+1$ bit-stuffing scheme with matrix representation C. If $|U|/N < 1/(j+1)$, then the average gap size is $> j$. This leads to the question of whether at insertion rates $< 1/(j+1)$, configurations with unconstrained positions spaced uniformly apart by the average gap size yield code rates that are close to optimal. We show below in the next proposition that this is not the case: for a fixed period and insertion rate $< 1/(j+1)$, it is possible to reduce all the large gaps (except one) to size exactly j, yielding improvements in code rates.

Proposition 7: Let $j \leq 7$. Let $g_1, \ldots, g_k$ be large gaps such that $g_1 \geq 3j$ and $g_i \geq j$ for $i = 2, \cdots, k$, and let $G_k = \sum_{i=1}^{k} g_i$. Then

$$C^k A^{G_k - kj} \succcurlyeq BA^{g_k} BA^{g_{k-1}} \cdots BA^{g_2} BA^{g_1}.$$

C. Unifying the Arguments on Large and Small Gaps

The proof of Theorem 3 involves applying the lemmas on large and small gaps alternately to a matrix $M_0 = BA^{g_1} \cdots BA^{g_k}$ with k gaps of length N, and whose insertion rate k/N is $< 1/(3(j+1))$. It will be shown in stages that there is a sequence of matrices $M_0, \ldots, M_q$ such that for $i = 1, \ldots, q$, there is a matrix M' such that $M_i \succcurlyeq M'$ and $\lambda(M') = \lambda(M_{i-1})$ and $M_q = C^k A^{G-kj}$. Each M_{i-1} will be a product of a sequence of A's and B's and M' will be a product of a cyclically permuted sequence of A's and B's.

Sketch of proof: [Theorem 3:] View the interval $\{1, \ldots, k\}$ as a sequence of uniformly spaced points on the unit circle, with numbers increasing to the right moving clockwise. Subdivide the circle into maximal, consecutive disjoint subintervals $I_1, \ldots, I_r$ such that for all odd (even) s and for all $i \in I_s$, g_i is large (small).

The average value of the g_i's is at least $3j$ because of the assumption on the insertion rate. For odd s, if $I_s = \{p, \ldots, q\}$ contains an index l (if there is more than one, choose the rightmost one) such that $g_l \geq 3j$, then applying Proposition 7 to the g_i's for $i = p, \ldots, l$ yields new $g_i = j$ for $i = p, \ldots, l-1$, $g_l \geq 3j + \sum_{i=p}^{l} g_i - (l-p)j$, and $g_i < 3j$ for $i = l+1, \ldots, q$. The resulting matrix product in this interval is thus

$$(BA^j)^{l-p} BA^{g_l} BA^{g_{l+1}} \cdots BA^{g_q}.$$

In the process, we have changed neither the sum nor the number of g_i's, so the average g_i remains the same.

Decompose the intervals into adjacent pairs, $I_s \cup I_{s+1}$ (where s is even and the subscripts are taken mod r). A weighted average over all s in $\{1, \ldots, r\}$ of $\sum_{i \in I_s \cup I_{s+1}} g_i / |I_s \cup I_{s+1}|$ gives the average g_i and hence $\geq 3j$, so we can choose a pair $I_s \cup I_{s+1}$ with indices $I_s = \{u, \ldots, v\}$ and $I_{s+1} = \{v+1, \ldots, w\}$ satisfying $\sum_{i=u}^{w} g_i \geq 3j(w-u+1)$.

As noted above, I_{s+1} contains an index l such that $g_l \geq 3j$. and $g_i < 3j$ for $i = l+1, \ldots, w$.

It is straightforward to check that Lemma 6 is applicable to I_s by showing that g_l has sufficient A's for topping up the small gaps in I_s. Applying Lemma 6 to $I_s \cup I_{s+1}$ thus results in replacing $BA^{g_u} \cdots BA^{g_l} BA^{g_{l+1}} \cdots BA^{g_{w-1}} BA^{g_w}$ by $C^{l-u} BA^{g_l'} BA^{g_{l+1}} \cdots BA^{g_w}$, where g_l' is the new value of g_l after topping up the small gaps in I_s.

Next we merge $I_s \cup I_{s+1}$ with $I_{s-1} = \{t, \ldots, u-1\}$, which also consists of large gaps. If $g_l' \geq 3j$, Proposition 7 can be applied to $I_{s+1} \cup I_s \cup I_{s-1}$ to drain the excess A's from each large gap in I_{s-1} into g_l'. Denote the new value of g_l' by g_l''. The merging process replaces $BA^{g_t} \cdots BA^{g_{u-1}} C^{l-u} BA^{g_l'} BA^{g_{l+1}} \cdots BA^{g_w}$ with $C^{l-t} BA^{g_l''} BA^{g_{l+1}} \cdots BA^{g_w}$. If $g_l' < 3j$, we can pick another pair of intervals $I_{s'} \cup I_{s'+1}$ (with s' is even, and $s' \neq s$) where there is a gap in $I_{s'+1}$ with size $\geq 3j$. Such a gap always exists because the size of the sum of all gaps does not change. The procedure of alternately applying Proposition 7 and Lemma 6 can be then performed repeatedly until there are no remaining gaps $g_i < j$. ■

It remains an open problem to extend these results to an arbitrary j.

References

[1] J. Campello de Souza, B. H. Marcus, R. New, and B. A. Wilson, "Constrained systems with unconstrained positions," *IEEE Trans. Inform. Theory*, vol. 48, no. 4, pp. 866–879, Apr. 2002.

[2] R. D. Cideciyan, E. Eleftheriou, B. H. Marcus, and D. S. Modha, "Maximum transition run codes for generalized partial response channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 619–634, Apr. 2001.

[3] K. A. S. Immink, "A practical method for approaching the channel capacity of constrained channels," *IEEE Trans. Inform. Theory*, vol. 43, no. 5, pp. 1389–1399, Sept. 1997.

[4] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Handbook of Coding Theory*. Amsterdam, The Netherlands: Elsevier, 1998, ch. 20.

[5] J. Moon and B. Brickner, "Maximum transition run codes for data storage systems," *IEEE Trans. Magnetics*, vol. 32, no. 5, pp. 3992–3994, Sept. 1996.

[6] T. L. Poo, "Optimal code rates for constrained systems with unconstrained positions: an approach to combining error correction codes with modulation codes for digital storage systems," Ph.D. dissertation, Stanford University, 2005.

[7] T. L. Poo, P. Chaichanavong, and B. H. Marcus, "Tradeoff functions for constrained systems with unconstrained positions," *will appear in IEEE Trans. Inform. Theory*, April 2006.

[8] A. J. van Wijngaarden and K. A. S. Immink, "Maximum runlength-limited codes with error control capabilities," *IEEE J. Select. Areas Commun.*, vol. 19, no. 4, pp. 602–611, Apr. 2001.